

On q -ary LDPC Code Design for a Low Error Floor

Andrea Marinoni¹, Pietro Savazzi¹, Richard D. Wesel²

¹ Università degli Studi di Pavia, Dipartimento di Elettronica, Via Ferrata 1, 27100, Pavia, Italy.

Email: {andrea.marinoni, pietro.savazzi}@unipv.it

² University of California, Los Angeles, Electrical Engineering Department, Los Angeles, CA 90095-1594, USA.

Email: wesel@ee.ucla.edu

Abstract—This paper explores protograph-based and ACE-based methods for constructing q -ary low-density parity-check (LDPC) matrices. The ACE approach maximizes approximate cycle extrinsic message degree, explicitly avoiding small q -ary stopping sets and implicitly avoiding small absorbing sets. In addition to ACE, this paper applies linear-dependent-set maximization (LDSM) to the binary image of the q -ary LDPC matrix. Performance is studied for binary and q -ary instances of erasure channels and additive white Gaussian noise channels. The combination of the ACE approach and LDSM provides dramatic error floor improvement for the binary erasure channel and both binary and q -ary AWGN channels.

I. INTRODUCTION

Low-density parity-check (LDPC) codes [1] approach the channel capacity. Specifically, proper degree distribution design [8], [9], [11] allows LDPC codes to approach the Shannon limit. For a specified error rate, designing and decoding the LDPC code over $GF(q)$ where $q > 2$ [2], [3], [16] can improve the LDPC code parameters over a $GF(2)$ design to allow increased code rate, reduced blocklength, or increased intersymbol interference (ISI) resilience. Such codes are called q -ary (or non-binary) LDPC codes, and this paper focuses on the design of these q -ary LDPC codes to achieve low error floors.

Recently LDPC codes have found application in emerging storage technologies such as Flash memories [19], [20] and Phase Change Memories (PCMs) [18]. These devices require very good error-rate performance at very high code rate. The readout channel model is typically memoryless and it can be either binary-input (e.g. for standard PCMs [18]) or q -ary-input (e.g., for multilevel cell PCMs [21]). Moreover, LDPC codes for storage technologies must have extremely low error floors. In that regard, q -ary LDPC codes that combine very low error floors and acceptable decoding complexity may play an important role in solid state device readout systems, since they typically provide very good performance in the waterfall region at high coding rate.

Starting from the baseline q -ary protograph approach introduced in [17], this paper constructs q -ary LDPC codes with improved error floors using algorithms that avoid the graphical structures that cause the error floor. Stopping sets and absorbing sets are well-known graphical structures that cause error floors in binary LDPC codes. This paper identifies the analogous structures for q -ary LDPC codes. For q -ary codes, linear dependent sets also cause error floor behavior even in cases where the linear dependent set is neither a stopping set or

an absorbing set. This paper introduces a linear-dependent-set maximization (LDSM) algorithm that specifically addresses the problem of small linear dependent sets.

To summarize, this paper provides efficient construction methods for q -ary LDPC codes with moderate field size to achieve low error floors. These algorithms improve performance by constructing LDPC matrices that avoid small q -ary stopping sets, absorbing sets, and linearly dependent sets. The proposed algorithms are studied for binary and q -ary instances of erasure, symmetric, and additive white Gaussian noise channels.

The paper demonstrates the importance of the channel when considering the LDPC matrix construction algorithm. The combination of the ACE approach and linear-dependent-set maximization provides dramatic error floor improvement for the binary erasure channel and both binary and q -ary AWGN channels. The improvements obtained with ACE and LDSM were less significant for q -ary erasure channel.

The paper is organized as follows: Section II describes the matrix representations for q -ary LDPC codes and presents the system model. Section III provides a graphical analysis of the q -ary decoding problem. Specifically, definitions, properties and relationships of q -ary cycle sets, q -ary stopping sets and linearly dependent sets are provided. Section IV provides the low error floor q -ary LDPC code construction algorithms. Section V presents the simulation results and discusses the practical aspects of the code construction and decoding. Section VI delivers the conclusions.

II. LDPC MATRIX REPRESENTATIONS & SYSTEM MODEL

This paper uses LDPC codes over $GF(q)$, with $q=2^p$ and p an integer. For code rate R , and block length N , an LDPC code is described by an $M \times N$ parity-check matrix, $H = \{H_{ij}\}_{i=1,\dots,M;j=1,\dots,N}$, where $H_{ij} \in GF(q)$ and $R = 1 - M/N$. For a specified LDPC matrix H , the binary mother matrix $\chi(H)$ and the binary image matrix $\psi(H)$ are defined as follows:

Definition 1 (Binary Mother Matrix): The **binary mother matrix** (BMM) $\chi(H)$ has the same dimensions as H and identifies the positions of the non-zero entries in H so that $\chi(H_{ij}) = 1$ if $H_{ij} \neq 0$ and $\chi(H_{ij}) = 0$ if $H_{ij} = 0$.

Definition 2 (Binary Image Matrix): The **binary image matrix** (BIM) $\psi(H)$ expands H into a $pM \times pN$ binary matrix [13]. Each element H_{ij} of H is represented in $\psi(H)$ by the $p \times p$ binary matrix that corresponds to that element.

To accomplish the expansion, let $a(x) = a_0 + a_1x + \dots + a_{p-1}x^{p-1} + x^p$ with $a_i \in GF(2)$ be the defining primitive polynomial of the Galois field $GF(q)$ where $q = 2^p$. Let A be the $p \times p$ matrix representation of the primitive element α of $GF(q)$. Thus, A is as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & 0 \\ 0 & 0 & 0 & \ddots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{p-1} \end{bmatrix}. \quad (1)$$

The powers of A correspond to the powers of α , i.e. the non-zero elements of $GF(q)$. That is, $\alpha^\kappa \Leftrightarrow A^\kappa$, where $\kappa \in \{0, \dots, q-2\}$. With this notation in hand, $\psi(H)$ is comprised of $p \times p$ sub-matrices $\psi(H_{ij})$ such that $\psi(H_{ij}) = A^t$ if $H_{ij} = \alpha^t$ and $\psi(H_{ij}) = \mathbf{0}$ if $H_{ij} = 0$, where $\mathbf{0}$ is the $p \times p$ all-zeros matrix.

$K = NR$ information symbols from $GF(q)$ are encoded into N coded symbols from $GF(q)$, which can be represented by a q -ary vector $\mathbf{x} = \{x_j\}_{j=1, \dots, N}$, $x_j \in GF(q)$. These coded symbols might be transmitted over a q -ary input channel or a binary input channel. For a q -ary-input channel, the N elements of \mathbf{x} are mapped directly into the modulated sequence $\mathbf{s} = \{s_j\}_{j=1, \dots, N}$. For a binary-input channel, the q -ary vector \mathbf{x} is mapped to the binary vector \mathbf{x}_b by replacing each q -ary element with the p bits that correspond to the binary representation of that $GF(q)$ element. Then each bit of \mathbf{x}_b is mapped directly into the modulated sequence $\mathbf{s} = \{s_j\}_{j=1, \dots, Np}$. We use \bar{x}_b^j to identify the vector of p bits in \mathbf{x}_b that correspond to the q -ary symbol x_j .

At the receiver, the channel outputs are properly demapped and the probabilities that are used to initialize the message passing algorithm in the decoder [2] are computed. Specifically, for a binary-input channel, the channel output is $\mathbf{y}_b = \{\bar{y}_b^j\}_{j=1, \dots, N}$, where $\bar{y}_b^j = \{y_{b_k}^j\}_{k=1, \dots, p}$ represents the set of p channel outputs corresponding to a single transmitted $GF(q)$ element represented by \bar{x}_b^j . Because the binary channel is memoryless, the conditional density of the j -th symbol x_j is $f_j(x_j = \beta | \bar{y}_b^j) = \prod_{k=1}^p P(x_{b_k}^j = \beta_k | y_{b_k}^j)$ for each $j = 1, \dots, N$, $\beta \in GF(q)$, where β_k is the k -th bit of the binary representation of β . For a q -ary-input channel where each y_j is associated with the j -th transmitted symbol x_j , the conditional density $f_j(x_j = \beta | y_j)$ can be properly computed directly from the respective channel output.

We define fully and partially erased symbols as follows:

Definition 3 (Fully Erased Symbol): A symbol is **fully erased** if all the bits that identify its binary representation are erased.

Definition 4 (Partially Erased Symbol): A symbol is **partially erased** if some but not all of the bits that identify its binary representation are erased.

III. DECODING FAILURES AND THE BIPARTITE GRAPH

The ACE algorithm [4] is an effective algorithm for lowering the error floors of irregular binary LDPC codes with a specified degree distribution. While explicitly designed to

avoid stopping sets [5], this algorithm also avoids absorbing sets [23] and trapping sets [24] which also require cycles with a relatively low cycle EMD.

It is well-known that the degree distribution plays an important role in LDPC code performance. It is possible to optimize both the degree distribution of the the BMM and the distribution of field elements as in [13], [14], [15]. However, optimizing the field-element distribution is computationally cumbersome, and prohibitive for field sizes larger than eight [14]. Still, careful selection of the non-zero field elements [12], [13], [17] in the q -ary parity-check (PC) matrix is crucial for good error floor performance.

As in [4], we aim to develop a construction algorithm based on a correspondence between decoder failures and graphical topologies. However, where [4] focused only on $GF(2)$ we are interested in higher Galois field sizes. To highlight the similarities and differences with their binary counterparts, we extend some definitions introduced in [4] to the q -ary case.

Definition 5 (q -ary Cycle): A set of d q -ary variable nodes and d q -ary constraint nodes forms a **q -ary cycle** if they are connected by edges induced by a BMM such that a path exists that travels through every node in the set and connects each node to itself without traversing a node twice.

Definition 6 (q -ary Cycle Set): A set of d q -ary variable nodes is a **q -ary cycle set** ξ_d if one or more q -ary cycles are formed between this set and its neighboring constraint set in a bipartite graph induced by the BMM $\chi(H)$.

Definition 7 (q -ary Stopping Set): A set of d q -ary code-word symbols is a **q -ary stopping set** S_d if no parity-check equation of H includes exactly one symbol in S_d .

The presence or absence of q -ary stopping sets depends entirely on the BMM $\chi(H)$ or equivalently its induced bipartite graph.

Lemma 1: If all d symbols in a q -ary stopping set S_d are erased, none of them can be recovered by message passing, even if all other symbols are received without error.

Proof: If all d symbols in S_d are erased, every parity check equation that involves a symbol in S_d has at least two erasures. Thus no parity check equation can correct an erased symbol in S_d . Hence message passing fails to correct any of the erased symbols in S_d when all are erased. ■

Lemma 2: A set of variable nodes such that when all are erased, none can be recovered by message passing is always a stopping set.

Proof: In a q -ary erasure channel, if an erased variable node cannot be recovered by message passing, all constraint nodes connected to it must be connected to at least one additional erased symbol. Thus every constraint node must connect either to no variable nodes in the set or more than one variable node in the set. Thus the set is a stopping set. ■

Remark. Definitions 5, 6, 7, Lemmas 1 and 2 are the natural extensions of the ideas in [4] and [5] to the q -ary case.

Definition 8 (Linearly Dependent Set): A **linearly dependent set** $\Lambda_{d,l}$ is a set of l binary columns of the BIM $\psi(H)$ that are linearly dependent. These l binary columns come from d q -ary columns of H . Note that $l \geq d$.

The relationships among q -ary cycle sets, q -ary stopping sets and linearly dependent sets provide insight into the structure of a given q -ary LDPC code. The following lemmas and theorems elaborate on these relationships between ξ_d , S_d , and $\Lambda_{d,l}$.

Lemma 3: In the bipartite graph induced by BMM $\chi(H)$ without singly connected variable nodes, every q -ary stopping set contains at least one q -ary cycle ($\{\xi_d\} \supset \{S_d\}$).

Proof: Since q -ary cycle sets and q -ary stopping sets are basically defined over the bipartite graph induced by a BMM, Lemma 1 can be proved as Lemma 1 in [4]. ■

Theorem 1: The d variable nodes that contain a linearly dependent set $\Lambda_{d,l}$ must form a stopping set S_d .

Proof: When the l bits corresponding to the l linearly dependent columns of $\psi(H)$ are erased, it is impossible to successfully decode any of these bits because two different codewords exist, one with all these bits equal to zero and one with all these bits equal to one. Since when all l of these bits are erased, none can be recovered, when all d corresponding q -ary symbols are erased, none can be recovered. Hence these d symbols form a stopping set by Lemma 2. ■

Corollary 1: For a q -ary LDPC code with no singly connected nodes, the d q -ary nodes that contain a linearly dependent set $\Lambda_{d,l}$ must also contain at least one cycle.

Proof: By Theorem 1, these d nodes form a stopping set. By Lemma 3, every stopping set contains a cycle. ■

Corollary 2: If d columns of the q -ary PC matrix sum to zero over $GF(q)$, these columns contain p ($p = \log_2 q$) linearly dependent sets $\Lambda_{d,l}$ with $l = d$.

Proof: Let D be the index set of the d columns of the q -ary PC matrix to be summed. The sum over $GF(q)$ of these columns is the all-zero vector. That is,

$$\sum_{j \in D} H_{ij} = 0 \quad (2)$$

in $GF(q)$ for all $i=1, \dots, M$. Recalling that $GF(q)$ addition corresponds to modulo-2 addition of the binary images, the columns of the binary image of the q -ary PC matrix columns associated with the d q -ary variable nodes have to sum to zero as well. Thus,

$$\bigoplus_{j \in D} \psi(H_{ij})_{tk} = 0 \quad (3)$$

for all $k \in \{1, \dots, p\}$, $t \in \{1, \dots, p\}$, and all $i \in \{1, \dots, M\}$. Therefore, the d q -ary variable nodes in D form p linearly dependent sets $\Lambda_{d,d}$, one for each of the p binary columns (indexed by k in (3)) of the binary image of a $GF(q)$ column. ■

Using the definitions and relationships of the topological structures established in this section, the next section connects these topological structures with corresponding decoding failures and introduces algorithms that exploit these correspondences to improve the error-rate performance of the q -ary LDPC codes.

IV. LOW ERROR FLOOR DESIGN

This section introduces two algorithms that mitigate the effects of the topological structures identified in the previous section. The combination of these algorithms guarantees that the smallest q -ary stopping sets and linearly dependent sets will be as large as possible for a given degree distribution.

Among the topological structures introduced in Section III, q -ary stopping sets and linearly dependent sets are the main contributors to the error-rate performance. The following two subsections introduce two algorithms that aim to counteract the effect of q -ary stopping sets and linearly dependent sets respectively. These algorithms are the q -ary ACE algorithm and the LDSM algorithm.

A. The q -ary ACE algorithm

A decoding failure occurs when all the variable nodes in a q -ary stopping set S_d are completely erased. Since each constraint node connected to S_d is connected to at least two variable nodes in S_d , and all the variable nodes in S_d are erased, no constraint node can help any variable node in S_d recover from erasure. Therefore, it is not possible for the decoder to converge to a codeword, and the decoder fails.

The occurrence of q -ary stopping sets is completely determined by the configuration of the BMM $\chi(H)$ associated with the q -ary LDPC code. As shown above, both q -ary stopping sets and linearly dependent sets must contain cycles. Hence, these graphical structures can be controlled by controlling the cycles that they must contain. Although introduced for binary LDPC codes, the ACE algorithm [4] represents an efficient method to construct the BMM so as to increase the minimum q -ary stopping set size by increasing the connectivity (the extrinsic message degree) of small cycles.

B. The LDSM algorithm

A second type of decoding error is caused by partial erasure of the q -ary variable nodes in a q -ary stopping set, as would occur when q -ary symbols are transmitted one bit at a time over a binary channel. Such partial erasures cause a decoding error when the q -ary partial erasures include all the bits that form a binary linearly dependent set in the binary image of the q -ary PC matrix. In such a case, at least two codewords share the largest a-posteriori probability, so that even a maximum-likelihood decoder cannot decode successfully.

In the context of a binary channel, the minimum binary Hamming distance between two codewords is the minimum number of linearly dependent columns in the binary image of the q -ary PC matrix. Moreover, a code with minimum binary Hamming distance l_{min} has at least one $\Lambda_{d,l_{min}}$ set but no $\Lambda_{d,l}$ sets where $l < l_{min}$. Thus increasing the size of the smallest linear dependent set should also increase the minimum distance of the code.

We have shown in Theorem 1 that all linearly dependent sets are contained in stopping sets. However, for a given size stopping set, the size of the smallest linearly dependent set it contains depends on the selection of the nonzero field elements for the columns corresponding to the variables in that stopping

```

initialize: begin
     $B = \emptyset; R = \emptyset; U = \emptyset; T = \{1\};$ 
end initialize

while  $|B| < N$  do
begin
    for  $k = 1$  to  $|T|$  do
    begin
         $R = \text{findcol}(B, T_k);$ 
        for  $n = 1$  to  $|R|$  do
        begin
             $B \leftarrow B \cup R_n;$ 
             $\{H_{iR_n}\}_{i \in G(R_n)} = \text{choose}(B, q);$ 
            if  $B = \{1, \dots, N\}$ 
                exit;
            end if
        end for
         $U \leftarrow U \cup R;$ 
    end for
     $T \leftarrow U;$ 
end while

function:  $R = \text{findcol}(B, T_k)$ 
     $R_i \in \{1, \dots, N\} \setminus B$  such that
     $c_1(\{\chi(H_{uT_k}) \oplus \chi(H_{uR_i})\}_{u=1, \dots, M})$  is minimized.
end function

```

Fig. 1. The Linear-Dependent-Set Maximization (LDSM) Algorithm.

set. After the ACE algorithm determines the positions of the non-zero entries in the q -ary PC matrix, the value of each non-zero $H_{ij} \in H$ must be chosen. This paper introduces the linearly dependent set maximization (LDSM) algorithm to choose the non-zero entries so as to maximize the size of the smallest linearly dependent sets $\Lambda_{d,l}$.

The LDSM algorithm is performed sequentially, column by column. Let N be the number of the q -ary variable nodes (columns) and M be the number of the q -ary check nodes (rows) of the q -ary LDPC code. Let H , $\chi(H)$ and $\psi(H)$ be the q -ary PC matrix, the binary mother matrix and the binary image of H respectively. Let $B = \{B_i\}_{i=1, \dots, |B|}$, $R = \{R_i\}_{i=1, \dots, |R|}$, $T = \{T_i\}_{i=1, \dots, |T|}$ and $U = \{U_i\}_{i=1, \dots, |U|}$ be sets of column indices, i.e., $B_i, R_i, T_i, U_i \in \{1, \dots, N\}$. Further, let $c_1(z)$ be the number of ones in the binary sequence z . Finally, let $G(t)$ be the set of the row indexes of the non-zero entries of the t -th column of the q -ary PC matrix, i.e., $G(t) = \{i \in \{1, \dots, M\} : \chi(H_{it}) = 1\}$. With this notation, Fig. 1 describes the LDSM algorithm.

For a given BMM, the LDSM algorithm assigns Galois field elements to the column that is the most connected to the previously assigned columns. The elements are assigned to maximize the size of the smallest binary linearly dependent sets in the BIM.

Specifically, the function $\{H_{iR_n}\}_{i \in G(R_n)} = \text{choose}(B, q)$ first computes the vector space spanned by the binary columns of the submatrix of the BIM comprised of the q -ary columns that have been assigned so far. Then, it selects the non-zero entries of the R_n column to maximize the number of pM -length vectors that are orthogonal to that vector space.

Avoiding small linearly dependent sets increases the binary

Hamming distance of a q -ary LDPC code. The LDSM algorithm aims to construct q -ary LDPC codes having linearly dependent sets $\Lambda_{d,l}$ where $d_{min} = \min\{d\}$ and $l_{min} = \min\{l\}$ are as large as possible.

The role of q -ary stopping sets and linearly dependent sets can be translated to q -ary and binary non-erasure scenarios. In those cases, variables with poor observation reliability are analogous to erasures. Thus, increasing the minimum q -ary stopping set size and the minimum linearly dependent set size should represent an effective method for generating q -ary LDPC codes suited for message-passing decoding even in AWGN. Moreover, by employing the ACE algorithm as previously described to avoid small q -ary stopping sets, small absorbing sets and trapping sets are also avoided.

V. SIMULATION RESULTS

This section presents simulation results for both binary-input channels and q -ary-input channels using q -ary LDPC codes. Each code has $(N, R, q) = (2500, 1/2, 16)$, where N is the number of symbols for each codeword, R is the code rate and q is the alphabet size. Specifically, 2500 16-ary symbols correspond to 10000 transmitted bits. The primitive polynomial of the considered 16-ary Galois Field is $a(x) = 1 + x + x^4$. In this paper we consider Quasi-Regular (QR) LDPC codes [7], [9]. Further, each code is characterized by an average column weight $t = 2.6$ in order to have a fair comparison in terms of error-rate performance. The maximum number of iterations of the decoding MP algorithm [2], [3] has been set to 50.

The 16-ary LDPC codes have been constructed in a modular manner. First, the binary mother matrix is defined by using one of the following algorithms: (i) protograph-based algorithm [17]; (ii) ACE algorithm [4]; (iii) protograph-based ACE algorithm in which the connections of the BMM are defined by using the ACE algorithm [4] constrained by the protograph that is considered..

For the protograph-based PC matrices the number of transmitted variable nodes of the protograph has been set to 10, while the number of check nodes of the protograph has been set to 5. The protograph is copied and permuted 250 times to produce the 2500-symbol LDPC codes. The ACE PC matrices and the protograph-based ACE PC matrices have been constructed by setting the d_{ACE} and η parameters [4] to 20 and 1 respectively. The value of η is strictly related to the average column weight that has been taken into account.

The non-zero entries of the q -ary PC matrices are then chosen by using the following algorithms: (a) random selection, where the non-zero elements are randomly selected from the non-zero elements in $GF(q)$; (b) careful selection, where the non-zero elements in each row of the q -ary PC matrix are selected by using the method introduced in [17]; (c) selection by the linearly dependent set maximization (LDSM) algorithm.

The simulations considered both binary-input channels and q -ary-input channels. Specifically, we considered the binary erasure channel (BEC) in Fig. 2, the binary AWGN channel (BAWGNC) in Fig. 3, the q -ary erasure channel (qEC) in

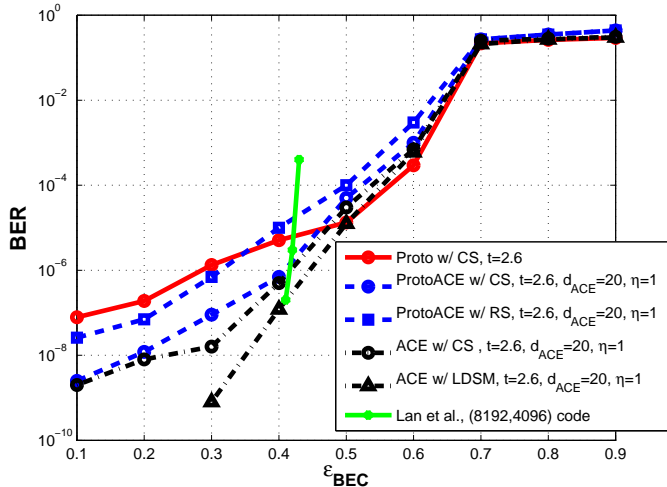


Fig. 2. Bit Error Rate performance of the considered q -ary LDPC codes on the BEC channel. The binary mother matrices are constructed by using protograph-based algorithm (“proto”) [17], ACE algorithm (“ACE”) [4] and protograph-based ACE algorithm (“protoACE”). The non-zero entries in the q -ary PC matrix are randomly selected (“RS”), carefully selected (“CS”) or selected by means of the LDSM algorithm (“LDSM”). The average column weight is t , whereas d_{ACE} and η are the setup parameters for the ACE algorithm. The abscissa label ε_{BEC} is the binary erasure probability. The code from Lan et al. is the best comparable binary LDPC code from [27].

Fig. 4, and the q -ary AWGN channel (q AWGNC) in Fig. 5. For the BAWGNC the signal is mapped onto a BPSK constellation, whereas a symmetric ultracomposite [22] Gray-labeled 16-QAM modulation is used for the q AWGNC ($q=16$).

The red solid line, the blue dashed line and the black dash-dot line represent an LDPC code whose BMM has been constructed by means of the protograph-based algorithm, the protograph-based ACE algorithm and the ACE algorithm respectively. The square marker, the circle marker, and triangle marker indicate an LDPC code whose non-zero entries in the q -ary PC matrix mother matrix are randomly selected, carefully selected and selected by using the LDSM algorithm respectively.

The code that has been constructed by means of the protograph-based algorithm in [17] shows the best performance in the waterfall region. However, it also shows the worst performance in terms of error floor on every channel that has been considered. The codes whose binary mother matrix has been constructed using the protograph-based ACE algorithm have similar error-rate performance in the waterfall region to those of the protograph-based q -ary LDPC code (except in the q -ary AWGN simulaton). Further, they show lower error floors than the protograph-based q -ary LDPC code. Specifically, the selection of the non-zero entries in the q -ary PC matrix delivered by the algorithm in [17] provides an effective improvement of the performance w.r.t. the random selection algorithm. This benefit is shown in the waterfall region as in the error floor region as well.

On the other hand, the q -ary LDPC codes whose binary mother matrix is constructed by means of the ACE algorithm

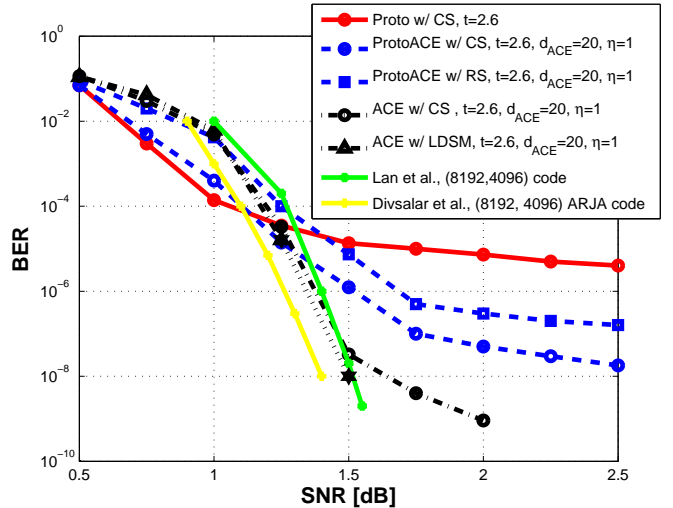


Fig. 3. Bit Error Rate performance of the considered q -ary LDPC codes on the binary AWGN channel. The codes are described in Fig. 2 except the codes from Lan et al. and Divsalar et al. which are binary LDPC codes from [27] and [26] respectively. Specifically, the codes in [26] are precoded LDPC codes. The marker indicated with a star is only an upper bound on the performance of ACE with LDSM at 1.5 dB because no errors have yet been observed after 10^{10} decoded bits.

show the best error floor performance over each considered channel. The error floors of these codes are significantly lower than those related to protograph-based ACE constructed codes. The protograph constraints induce a sort of “super-structure” on the bipartite graph. This limits the q -ary stopping set mitigation of the protograph-based ACE algorithm so that it is not be as effective as that delivered by the standard ACE algorithm for quasi-regular and irregular codes.

Further, the selection of the non-zero entries performed by using the LDSM algorithm provides improvement in the error floor region for every channel that has been studied w.r.t. the selection algorithm in [17]. The gain provided by LDSM algorithm is substantial for the BEC, binary AWGN channel and q -ary AWGN channel, but not significant for the q -ary erasure channel. This is expected since the q -ary erasure channel has no partial erasures, which are the target of the LDSM algorithm.

VI. CONCLUSIONS

This paper provides a construction method for q -ary LDPC codes based on a careful geometrical and graphical analysis. For $GF(16)$, code construction examples and simulations are provided. Avoiding small q -ary stopping sets through a q -ary version of the ACE algorithm and linearly dependent sets through the new LDSM algorithm provided substantial improvement in the error floor over other construction algorithms. Simulation results over BEC and binary AWGN channel are very interesting as those channels can model the readout channel of emerging storage technologies [18], [21].

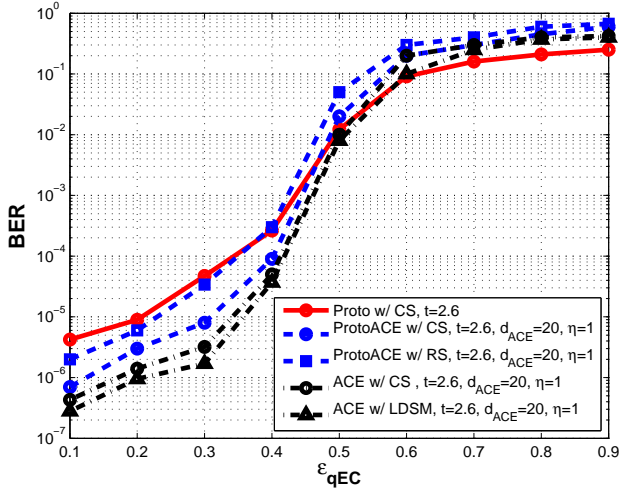


Fig. 4. Bit Error Rate performance of the considered q -ary LDPC codes on the q -ary erasure channel. The codes are described in Fig. 2. The abscissa label ϵ_{qEC} is the q -ary erasure probability.

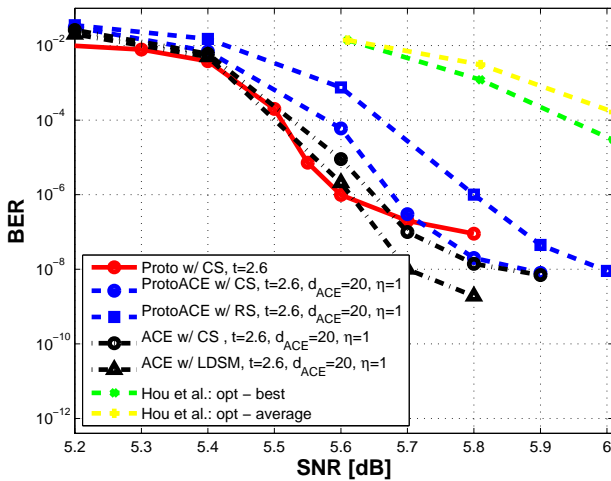


Fig. 5. Bit Error Rate performance of the considered q -ary LDPC codes on the q -ary AWGN channel ($q=16$ with 16-QAM modulation). The codes are described in Fig. 2 except the codes from Hou et al. which are the best comparable binary LDPC codes from [25].

REFERENCES

- [1] R.G. Gallager, *Low-Density Parity-Check Codes*, Cambridge, MA, MIT Press, 1963.
- [2] M.C. Davey, D. MacKay, "Low-Density Parity-Check Codes over GF(q)," *IEEE Communications Letters*, vol. 2, no. 6, June 1998.
- [3] M.C. Davey, "Error-Correction Using Low-Density Parity-Check Codes," Ph.D. Thesis, University of Cambridge, UK, December 1999.
- [4] T. Tian, C. Jones, J.D. Villaseñor, R.D. Wesel, "Construction of irregular LDPC codes with low error floors," in proc. *IEEE International Conference on Communications (ICC) 2003*
- [5] C. Di, D. Proietti, E. Telatar, T. Richardson, R. Urbanke, "Finite length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. on Inform. Theory*, vol. 48, pp. 1570-1579, June 2002.
- [6] N. Wiberg, "Code and decoding on general graphs," Dissertation no. 440, Dept. Elec. Eng., Linköping Univ., Linköping, Sweden, 1996.

- [7] B. Rong, T. Jiang, X. Li, M.R. Soleymani, "Combine LDPC Codes Over GF(q) With q -ary Modulations for Bandwidth Efficient Transmission," *IEEE Trans. on Broadcasting*, vol. 54, no. 1, March 2008.
- [8] T. Richardson, A. Shokrollahi, R. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, Feb. 2001.
- [9] T. J. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599-618, Feb. 2001.
- [10] J. Thorpe, "Low-Density Parity-Check (LDPC) Codes constructed from Protographs," JPL IPN Progress Report 42-154, Aug. 15, 2003.
- [11] S.Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D dissertation, Massachusetts Institute of Technology, Cambridge, MA, Sept. 2000.
- [12] D. MacKay, "Optimizing Sparse Graph Codes over GF(q)," available at <http://www.inference.phy.ca.ac.uk/mackay/CodesGallager.html>, Aug. 2003.
- [13] C. Poulliat, M. Fossorier, D. Declercq, "Design of regular $(2, d_c)$ -LDPC codes over GF(q) using their binary images," *IEEE Trans. Comm.*, vol. 56, pp. 1626-1635, Oct. 2008.
- [14] V. Rathi, R. Urbanke, "Density evolution, thresholds and the stability condition for non-binary LDPC codes," in *IEE Proceedings - Communications*, vol. 152, num. 6, 2005, p. 1069-1074
- [15] V. Rathi, "Conditional entropy of non-binary LDPC codes over the BEC," in proc. *IEEE International Symposium on Information Theory (ISIT) 2008*, Toronto, CN, July 6-11, 2008.
- [16] A. Marinoni, P. Savazzi, S. Valle, "Efficient Design of Non-Binary LDPC Codes for Magnetic Recording Channels, Robust to Error Bursts," in proc. *5th International Symposium on Turbo Codes and Related Topics*, Lausanne, CH, Sept. 1-5, 2008.
- [17] A. Marinoni, P. Savazzi, R.D. Wesel, "Protograph-based q -ary LDPC codes for higher-order modulation," in proc. *6th International Symposium on Turbo Codes and Related Topics*, Brest, F, Sept. 6-10, 2010.
- [18] A. Marinoni, A. Cabrini, E. Costamagna, G. Torelli, P. Gamba, "Challenges and opportunities for information theory-based design of Phase Change Memories," in proc. *IEEE Global Communications Conference (GLOBECOM) 2010*, Miami, FL, Dec. 6-10, 2010.
- [19] J. Wang, T. Courtade, H. Shankar, R. Wesel, "Soft Information for LDPC Decoding in Flash: Mutual-Information-Optimized Quantization," in proc. *IEEE Global Communications Conference (GLOBECOM) 2011*, Houston, TX, Dec. 5-9, 2011.
- [20] G. Dong, N. Xie, T. Zhang, "On the Use of Soft-Decision Error-Correcting Codes in NAND Flash Memory," *IEEE Trans. Circ. and Sys.*, 58(2):429439, Feb. 2011.
- [21] A. Pantazi, A. Sebastian, N. Papandreou, M.J. Breitwisch, C. Lam, H. Pozidis, E. Eleftheriou, "Multilevel Phase-Change Memory modeling and experimental characterization," in proc. *EPCOS 2009*
- [22] R.D. Wesel, X. Liu, C. Komninakis, and J.M. Cioffi, "Constellation Labeling for Linear Encoders," *IEEE Trans. on Inform. Theory*, vol. 47, no. 6, Sept. 2001, pp. 1572-1581.
- [23] L. Dolecek, Z. Zhang, M. J. Wainwright, V. Anantharam, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 1, Jan. 2010
- [24] T. Richardson, "Error-floors of LDPC codes," *Proc. 41st Annual Allerton Conf.*, Monticello, IL, Oct. 2003.
- [25] J. Hou, P.H. Siegel, L.B. Milstein, H.D. Pfister, "Capacity-Approaching Bandwidth-Efficient Coded Modulation Schemes Based on Low-Density Parity-Check Codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 9, Sept. 2003.
- [26] D. Divsalar, C. Jones, S. Dolinar, J. Thorpe, "Protograph based LDPC codes with minimum distance linearly growing with block size," in proc. *IEEE Global Communications Conference (GLOBECOM) 2005*, St. Louis, MO, Nov. 2005.
- [27] L. Lan, L. Zeng, Y.Y. Tai, L. Chen, S. Lin, K. Abdel-Ghaffar, "Construction of Quasi-Cyclic LDPC codes for AWGN and Binary Erasure Channel: a finite field approach," *IEEE Trans. Inform. Theory*, vol. 53, no. 7, July 2007.